**Instituto Tecnológico de Tijuana**

**Ingeniería en Sistemas Computacionales** 

**Examen II**

Machine Learning

**Materia:** Datos Masivos

**Unidad:** Unidad I

**Facilitador:**

José Christian Romero Sánchez

**Alumnos:**

* Hernández Negrete Juan Carlos - **16212021**
* Manzano Guzman Jesua - **16212033**

**Fecha:**

Tijuana Baja California a 15 de Diciembre del 2020.

**Introduction.**

For this evaluative practice, the tools to be used with Spark will be written, such as the MLLIB library, which is a distributed computing framework that allows the use of a series of Machine Learning algorithms, some methods and calculations to be performed, same as They deal with machine language (Machine Learning).

**Developing.**

**EVALUATION PRACTICE UNIT 2**

We use the Spark Mllib library for Machine Learning corresponding to multilayer perceptron

|  |
| --- |
| import org.apache.spark.ml.classification.MultilayerPerceptronClassifier import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator import org.apache.spark.sql.SparkSession import org.apache.spark.ml.feature.StringIndexer import org.apache.spark.ml.feature.VectorAssembler |

The object and the session are created to start spark and load the scala file

|  |
| --- |
| object Evaluation {   def main(): Unit = {  val spark = SparkSession.builder.appName("Evaluation").getOrCreate() |

1. The CSV file is imported to save the data in the dataframe as "df"

|  |
| --- |
| val df = spark.read.option("header", "true").option("inferSchema","true")csv("iris.csv") |

2. What is the name of the columns

|  |
| --- |
| df.columns |

3. How is the scheme?

|  |
| --- |
| val esquema = df.printSchema  df.printSchema |

4. Print the first 5 columns.

|  |
| --- |
| df.head(5) |

5. Use the describe () method to learn more about the data in the DataFrame.

|  |
| --- |
| val desc = df.describe().show()  df.describe().show() |

6. Make the pertinent transformation for the categorical data which will be our labels to be classified.

|  |
| --- |
| val labelIndexer = new StringIndexer().setInputCol("species").setOutputCol("indexedLabel").fit(df)  val indexed = labelIndexer.transform(df).drop("species").withColumnRenamed("indexedLabel", "label")  indexed.describe().show()  val assembler = new VectorAssembler().setInputCols(Array("sepal\_length","sepal\_width","petal\_length","petal\_width")).setOutputCol("features")  val features = assembler.transform(indexed)  val labelIndexer2 = new StringIndexer().setInputCol("label").setOutputCol("indexedLabel").fit(indexed)    features.show |

7. Build the classification model and explain its architecture.

// The data is divided into training (60%) and tests (40%)

|  |
| --- |
| val splits = features.randomSplit(Array(0.6, 0.4), seed = 1234L)  val train = splits(0)  val test = splits(1) |

// The layers of the neural network are specified:

// The input layer is size 4 (features), two layers in between

// one size 7 and the other size 7

// and 3 out (the classes)

|  |
| --- |
| val layers = Array[Int](4, 7, 7, 3) |

// Test set accuracy = 0.9607843137254902 (4, 5, 4, 3), (4, 8, 8, 3)

// Test set accuracy = 0.9803921568627451 (4, 9, 9, 3), (4, 7, 7, 3)

// Training parameters are established

|  |
| --- |
| val trainer = new MultilayerPerceptronClassifier().setLayers(layers).setBlockSize(128).setSeed(1234L).setMaxIter(100) |

// The model is trained

|  |
| --- |
| val model = trainer.fit(train)  val result = model.transform(test) |

8. Print model results

|  |
| --- |
| val predictionAndLabels = result.select("prediction", "label")  predictionAndLabels.show()   val evaluator = new MulticlassClassificationEvaluator().setMetricName("accuracy")  println(s"Test set accuracy = ${evaluator.evaluate(predictionAndLabels)}") |

**Conclusión**

All the topics seen in class were implemented, in addition to having investigated on the Spark page about the MLIB libraries and how they worked, how the layer architecture or neural network was built and thus being able to obtain the accuracy and precision, by conclusion, it is assumed by learned the topics.

**Repository**

[**https://github.com/JesuaMG/BigData/tree/Unit\_2/Unit2/Evaluation**](https://github.com/JesuaMG/BigData/tree/Unit_2/Unit2/Evaluation)

**Video**

[**https://youtu.be/v6qoozqW50Y**](https://youtu.be/v6qoozqW50Y)